Camera Relocalization with Ellipsoidal Abstraction of Objects

Vincent Gaudillière, Gilles Simon, Marie-Odile Berger

Camera Relocalization with Ellipsoidal Abstraction of Objects

Vincent Gaudillière* Gilles Simon* Marie-Odile Berger*

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Abstract

We are interested in AR applications which take place in man-made GPS-denied environments, as industrial or indoor scenes[1]. In such environments, relocalization may fail due to repeated patterns and large changes in appearance which occur even for small changes in viewpoint. We investigate in this paper a new method for relocalization which operates at the level of objects and takes advantage of the impressive progress realized in object detection. Recent works have opened the way towards object oriented reconstruction from elliptic approximation of objects detected in images. We go one step further and propose a new method for pose computation based on ellipse/ellipsoid correspondences. We consider in this paper the practical common case where an initial guess of the rotation matrix of the pose is known, for instance with an inertial sensor or from the estimation of orthogonal vanishing points. Our contributions are twofold: we prove that a closed form estimate of the translation can be computed from one ellipse-ellipsoid correspondence. The accuracy of the method is assessed on the LINEMOD database using only one correspondence. Second, we prove the effectiveness of the method on real scenes from a set of object detections generated by YOLO. A robust framework that is able to choose the best set of hypotheses is proposed and is based on an appropriate estimation of the reprojection error of ellipsoids. Globally, considering pose at the level of object allows us to avoid common failures due to repeated structures. In addition, due to the small combinatory induced by object correspondences, our method is well suited to fast rough localization even in large environments.

Index Terms: I.4.8 [Scene Analysis]: Tracking—; I.4.0 [General]: Image Processing Software; Computing methodologies—Computer graphics—Graphics systems and interfaces —Mixed / augmented reality ;

1 Introduction

Pose computation is one if not the most important problem of augmented reality. We are interested in AR applications which take place in man-made GPS-denied environments, as industrial or indoor scenes. In these environments, computer vision is often used to estimate the 6-DOF camera pose, rather than physical position or motion sensors, because it does not require any special equipment other than a camera. Moreover, vision-based techniques are likely to provide more accurate alignments of the virtual world with the real one, the latter being directly observed in the processed images.

Traditional approaches are based on matching of features, most often points, between images acquired in a pre-processing phase and images acquired during the use phase. When the 3D counterpart of these features is known, the pose can be inferred from the obtained 2D-3D correspondences by solving the classical PnP problem [9,15]. Unfortunately, the matching step generally relies on local image descriptors that, whether hand-crafted [19] or learned [37], are not robust to strong viewpoint or illumination changes [29]. Above all, different physical points of the scene may have a close local appearance. Adding these two problems sometimes leads to a situation where the ratio between the number of correct and incorrect matches is too low to compute the pose reliably, even using a robust procedure such as RANSAC [5].

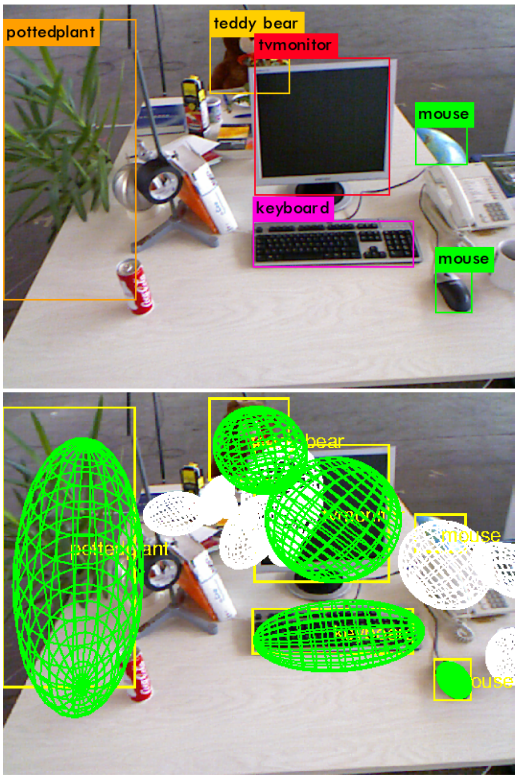

Figure 1: Our system is able to perform camera relocalization from objects detected in the image. Object detections (top) are modeled by ellipses, considered as projections of a known 3D scene model composed of 3D ellipsoids. The ellipsoids, once matched to 2D ellipses by our method, are reprojected in the image (bottom) using our estimated camera pose (green: inliers).

Recent works have shown that better performance can be achieved by training a convolutional neural network (CNN) to recognize either specific points (so-called control points) defined on an object model [2] or the vertices of a 3D bounding box (BB) of the object [13,23,34]. These methods take into account the overall appearance of the object and are even able, under certain conditions, to detect points occluded by another object. Unfortunately, a specific training of the CNN must be performed for each control point or vertex, using many images showing its appearance under various viewpoints, illumination, occlusion and background conditions. Moreover, the resulting network is only relevant for a specific instance of an object category, which makes this kind of methods not very portable e.g. to a new working environment.

By contrast, impressive progress have been made over the last few years in object class detection, thanks to methods based on CNNs such as R-CNN [7,10,27], SSD [18], or YOLO [24–26]. This qualitative leap has led to the emergence of new approaches to solving tradi-

*e-mail:firstname.lastname@inria.fr

[1]This work was funded by the DGA/DGE RAPID EVORA Project.

tional computer vision problems based on high-level primitives (objects), instead of the traditional low-level primitives [16, 17, 22, 28]. However, it is not easy to accurately compute the camera pose from coarse 2D BBs of objects, particularly when few objects are detected. For example, Li et al. use detection of BBs from two far-apart viewpoints to predict the 6-dof camera transformation between views [16]. Objects are modeled by their bounding cuboids and detected by using Faster-RCNN [27]. However, as the perspective projection of a cuboid is not a 2D box, a brute force, discrete optimization scheme has to be used to estimate the object poses, from a large set of hypotheses aligning well with the associated detection BB. A contextual model (vector describing some spatial relationships between the cuboids) is used to select the hypotheses that best ensure the global consistency, but this model has to be constructed manually, which, again, does not make it easy to adapt this system to a new environment.

Ellipsoidal abstraction of objects provides a more interesting geometric framework. An ellipsoid projects into an ellipse, these two primitives having a single equation to define them and well-established properties e.g. in the dual 3D and (resp.) 2D projective spaces. This paves the way for elegant and efficient solutions to align reprojected ellipsoids with the associated ellipses. Thereby in [28], Rubino *et al.* show how to automatically estimate an ellipsoid in 3D given a set of ellipses fitted to the object detection BBs in multiple calibrated views. The authors show that a closed-form solution exists from three views. In [22], Nicholson *et al.* describe a complete SLAM (Simultaneous Localization And Mapping) system that is able to reconstruct ellipsoids in 3D and simultaneously compute the camera poses from several ellipses fitted, again, to the object detection BBs. YOLOv3 [26] is used for object detection. A geometric error is minimized using a nonlinear iterative optimization process. However, this optimization is initialized by, and constrained to stay close to, some odometry measurements, which limits the scope of this method.

In this paper, we build on the advances made by Rubino et al. [28] and Nicholson et al. [22] to design a robust, portable and potentially sensor-less method that estimates the pose of a camera at object level (see Fig. 1). Our main contributions are the following.

First, we prove that a closed form estimate of the translation can be computed from a single ellipse-ellipsoid correspondence and known camera orientation (Section 3). This theoretical result has a great practical interest. Indeed, if capturing a camera orientation from external data (IMU, vanishing points, etc.) is quite easy, capturing a camera position is more difficult: odometry is known to suffer from cumulative drift, GPS is unusable indoors, and outside-in tracking (optical, WIFI, etc.) requires special equipment together with a tedious hand-eye calibration preprocess. Robustness of the camera position estimate against camera orientation errors and ellipse detection errors is assessed using the LINEMOD dataset [11] (Section 5.1).

Second, a robust framework able to choose the best hypotheses in presence of erroneous matches is proposed, which is based on an appropriate estimation of the reprojection error of ellipsoids (Section 4). Object class detection is known to be relatively safe but, of course, detection errors and / or misclassifications can still occur. Most importantly, if object class detection is interesting in terms of portability (the trained CNNs are relevant for any instance of the object class), as a counterpart, knowing the labels of the BBs does not allow distinguishing between several objects corresponding to the same class, so that each hypothesis has to be considered in the pose estimation process. Fortunately, this problem is somewhat offset by the fact that, knowing the camera orientation, its position can be inferred from only one ellipse-ellipsoid correspondence. Contrary to the PnP problem, one ellipse-ellipsoid correspondence brings information on a whole object and encompasses many local feature correspondences. As shown in our experiments, estimation from one correspondence is usually robust but the accuracy may depend on the accuracy of the detected ellipse. In order to cope with possible recognition errors and erroneous image/model association when several occurrences of an object are present in the scene, we propose a robust estimation framework with a low combinatorial cost which takes into account the projections of the other ellipsoids.

Finally, we prove the effectiveness of the method for camera relocalization in two sequences of the TUM RGB-D dataset [33] (Section 5.2). The sets of possible correspondences are generated by using YOLOv3, and camera orientations are provided by automatic extraction of vanishing points or simulated IMU data.

2 More related work

A strong limitation of the method described in [16], in addition to the high dimensionality of the 3D cuboid search space is that the scale of the selected objects is known, and the camera's viewing angle with respect to the ground plane is fixed. For sake of generality, these constraints are removed in [17], by inferring vanishing points from each image, and taking a short video, 25 consecutive frames from each viewpoint, allowing relative depth to be recovered for keypoint pixels using Structure From Motion (SFM). However, these additional steps make the method more cumbersome, especially since several hypotheses are still considered for each cuboid, before reasoning about object correspondences using the Hungarian algorithm.

Modeling object projections by virtual ellipses allowed Crocco *et al.* to propose a closed-form solution for SFM reconstruction of the scene in the form of an ellipsoid cloud [3]. However, this method is limited to the case of an orthographic projection, as well as its extension integrating CAD object models for higher reconstruction accuracy [6].

The case of perspective projection is considered in [22]. However, as we mentioned in the introduction, the SLAM resolution is constrained by odometry measurements. More precisely, the maximum a posteriori configuration of camera poses X^* and dual quadrics Q^* is found by solving the following non linear least squares problem:

$$X^*, Q^* = \text{argmin}_{X,Q} \sum_i ||f(\mathbf{x}_i, \mathbf{u}_i) \ominus \mathbf{x}_{i+1}||^2_{\Sigma_i} + \sum_{ij} ||\mathbf{b}_{ij} - \beta_{(\mathbf{x}_i, \mathbf{q}_j)}||^2_{\Lambda_{ij}}. \quad (1)$$

The first term reflects the attachment to odometry measurements and the second term is a geometric error between the projected ellipsoids and the detection BBs (see [22] for the details of the notations). The geometric error is defined as the sum of squares of the distances between borders of the BBs of the reprojected ellipsoids and borders of the detection BBs. Unfortunately, we show in Section 5.2 that removing the first term of the cost function (attachment to odometry), fixing the ellipsoid parameters $\mathbf{q}_j$ (shape and pose in the world frame of the ellipsoids) and minimizing the geometric term over the six pose parameters $\mathbf{x}_i$ can cause the optimization process to diverge severely, even with ground truth as initialization parameters. This is one reason why we argue that it is better to compute the camera orientation separately, and only estimate the camera translation (in closed-form) by using the ellipse-ellipsoid correspondences. Another limitation of this method, is that the associations between individual detections and distinct physical objects are provided by a set of manual annotations, while we propose a robust procedure to automatically determine the correct associations.

Actually, a closed-form solution for pose computation from ellipse-ellipsoid correspondences was first proposed in [35, 36], but only for the special case of spheroids (ellipsoid with two equal semi-diameters). In that case, the authors showed that, considering perspective projection, the spheroid pose estimation problem admits only two distinct solutions. In the more general case of ellipsoids, an equation of the same problem was proposed by Eberly [4] (which is the starting point of our theoretical development) without, however, an explicit method for calculating solutions.

3 CAMERA POSE COMPUTATION FROM ONE ELLIPSE - ELLIPSOID CORRESPONDENCE

We focus on the problem of camera pose estimation from one ellipse - ellipsoid correspondence. We consider the equivalent problem that consists in calculating the ellipsoid pose in the camera frame, and show that the ellipsoid position can be inferred from its orientation unambiguously.

3.1 The Cone Alignment Equation

Unless otherwise stated, all the variables introduced below are expressed in the camera coordinate frame.

Following the notations introduced in [4] and presented in Fig. 2, we consider an ellipsoid defined by

$$(\mathbf{X}-\mathbf{C})^\top A(\mathbf{X}-\mathbf{C}) = 1$$

where $\mathbf{C}$ is the center of the ellipsoid, A is a real positive definite matrix characterizing its orientation and size, and $\mathbf{X}$ is any point on it.

Given a center of projection $\mathbf{E}$ and a projection plane of normal $\mathbf{N}$ which does not contain $\mathbf{E}$, the projection of the ellipsoid is an ellipse of center $\mathbf{K}$ and of semi-diameters a et b. Ellipse's principal directions are represented by unit-length vectors $\mathbf{U}$ and $\mathbf{V}$, such that $\{\mathbf{U},\mathbf{V},\mathbf{N}\}$ is an orthonormal set.

The goal of this section is to compute the ellisoid position C given E, A and the detected ellipse on the image plane.

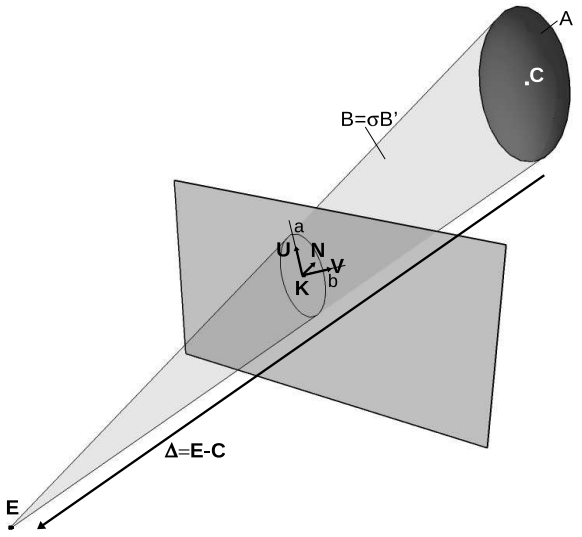

Figure 2: Illustrating the projection plane, projection center, ellipsoid and ellipse projection.

3.1.1 Projection Cone

The "projection cone" refers to the cone of vertex $\mathbf{E}$ tangent to the ellipsoid. According to [4], it is defined by the matrix

$$B \stackrel{def}{=} A\mathbf{\Delta}\mathbf{\Delta}^\top A - (\mathbf{\Delta}^\top A\mathbf{\Delta} - 1)A$$

where $\mathbf{\Delta} = \mathbf{E}-\mathbf{C}$, so that the points $\mathbf{X}$ on the projection cone are those who satisfy the equation $(\mathbf{X}-\mathbf{E})^\top B(\mathbf{X}-\mathbf{E}) = 0$.

3.1.2 Backprojection Cone

The "backprojection cone" refers to the cone generated by the lines passing through $\mathbf{E}$ and any point on the ellipse. Eberly shows that such a cone is characterized by the matrix B' defined as follows

$$B' \stackrel{def}{=} P^\top MP - Q$$

where

$$M \stackrel{def}{=} \mathbf{U}\mathbf{U}^\top/a^2 + \mathbf{V}\mathbf{V}^\top/b^2$$
$$\mathbf{W} \stackrel{def}{=} \mathbf{N}/(\mathbf{N}\cdot(\mathbf{K}-\mathbf{E}))$$
$$P \stackrel{def}{=} I - (\mathbf{K}-\mathbf{E})\mathbf{W}^\top$$
$$Q \stackrel{def}{=} \mathbf{W}\mathbf{W}^\top$$

Here again, the points $\mathbf{X}$ on the backprojection cone are those who meet $(\mathbf{X}-\mathbf{E})^\top B'(\mathbf{X}-\mathbf{E}) = 0$.

3.1.3 Alignment Equation

The ellipsoid projects on the ellipse if and only if the projection and backprojection cones are aligned [4], *i.e.* if and only if there is a non-zero scalar σ such that $B = \sigma B'$:

$$A\mathbf{\Delta}\mathbf{\Delta}^\top A - (\mathbf{\Delta}^\top A\mathbf{\Delta} - 1)A = \sigma B' \tag{2}$$

3.2 Computing position from orientation

Assuming that both size and orientation of the ellipsoid are known, the matrix A is fully determined. In the following we explain how to compute the remaining unknowns $\mathbf{\Delta}$ and σ of (2) from A and B'.

First, we will show that equation (2) implies

$$A\mathbf{\Delta} = \sigma B'\mathbf{\Delta} \tag{3}$$

Proof. Right-multiplying (2) by $\mathbf{\Delta}$, we have

$$(A\mathbf{\Delta}\mathbf{\Delta}^\top A - (\mathbf{\Delta}^\top A\mathbf{\Delta} - 1)A)\mathbf{\Delta} = \sigma B'\mathbf{\Delta}$$

Since $\mathbf{\Delta}$ is a 3D vector, $\mathbf{\Delta}^\top A\mathbf{\Delta}$ is a scalar, thus

$$\begin{aligned}(A\mathbf{\Delta}\mathbf{\Delta}^\top A - (\mathbf{\Delta}^\top A\mathbf{\Delta} - 1)A)\mathbf{\Delta} &= A\mathbf{\Delta}(\mathbf{\Delta}^\top A\mathbf{\Delta}) - \mathbf{\Delta}^\top A\mathbf{\Delta}A\mathbf{\Delta} + A\mathbf{\Delta} \\ &= (\mathbf{\Delta}^\top A\mathbf{\Delta})A\mathbf{\Delta} - \mathbf{\Delta}^\top A\mathbf{\Delta}A\mathbf{\Delta} + A\mathbf{\Delta} \\ &= A\mathbf{\Delta}\end{aligned}$$

□

$\mathbf{\Delta}$ is thus a *generalized eigenvector* of the couple $\{A, B'\}$.

According to the properties of ellipsoids and cones, it can be shown that (see proof in Appendix A) :

Theorem 1.

- *the couple $\{A, B'\}$ has exactly two distinct real generalized eigenvalues, one of multiplicity 1 denoted σ_1 , and one of multiplicity 2 denoted σ_2.*
- *only σ_1 is solution of (2).*

As a result, σ can be uniquely determined from the ellipsoid orientation and from the ellipse detected in the image, by considering the eigenvalue of multiplicity 1. $\mathbf{\Delta}$ is thus proportional to the eigenvector $\mathbf{\Delta}_1$ with norm 1 associated with σ_1.

We thus have $\mathbf{\Delta} = \vec{\mathbf{CE}} = k\mathbf{\Delta}_1$. Substituting this expression in (2) leads to

$$k^2(A\mathbf{\Delta}_1\mathbf{\Delta}_1^\top A - \mathbf{\Delta}_1^\top A\mathbf{\Delta}_1 A) = \sigma_1 B' - A$$

In practice, due to uncertainties in the detection, a least mean square estimation of k^2 is computed. The only possible k is the one that allows the center of the ellipsoid to be in front of the camera (chirality constraint).

Finally, let us note that all the computations were made in the camera coordinate frame. Let ${}^cR_w$ be the rotation matrix from world coordinates to camera coordinates. In practice, this matrix is estimated from sensors of from vanishing points. If A_w characterizes

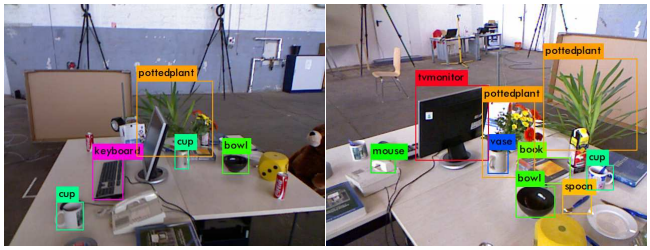

Figure 3: Illustration on the RGB-D TUM dataset [33] of the situation where a single object (the cup containing flowers in the center of the images) receive two different labels from the generic object detection algorithm YOLOv3 [26]: *cup* (left) and *vase* (right). These images also illustrate the fact that different objects (cups and plants) can be characterized by the same label: *cup* and *pottedplant*.

the size and orientation of the ellipsoid expressed in the world frame, then $A = {}^{c}R_{w} A_{w} {}^{w}R_{c}$ is the matrix expressed in the camera frame. From what precedes, we can then compute $\mathbf{\Delta} = \vec{\mathbf{EC}}$ expressed in the camera frame. Its expression in the world frame is thus given by ${}^{w}R_{c}\mathbf{\Delta}$. As the position of the ellipsoid is known in the world frame, the camera position can be determined.

4 Robust pose estimation method

A list a 2D/3D object association is the input of our pose procedure. Due to recognition errors, some of them may be wrong. In addition, as recognition systems are mostly learned on categories of object, there may be ambiguities in the choice of the physical object the detection originates from. The frequent occurrence of repeated objects in man-made environments is another source of false data association. In previous works [22], association was realized manually to avoid this problem. On the contrary, we propose a RANSAC-like procedure dedicated to robust 2D/3D object association with the aim to automatically discard false associations.

4.1 Model construction

Our camera relocalization system requires the knowledge of a light 3D model composed of ellipsoidal abstractions of objects of interest present in the scene. To reconstruct each object, possibly separately, a few frames with known poses (3 at minimum [28]) covering the widest possible range of viewpoints is necessary. In these frames, objects are automatically detected (in the form of rectangular bounding boxes) and labelled using an object detection algorithm (*e.g.* [10, 18, 26]), then the association between 2D detections across images is carried out manually. For each object, ellipses inscribed in detected bounding boxes are considered as 2D projections of an underlying 3D ellipsoid, therefore reconstructed using [28]. The labels are automatically transferred from 2D detections to 3D ellipsoids, sometimes leading to situations where a single object is described by several labels, and where different objects are described by the same label (see Figure 3).

4.2 RANSAC-like procedure for position estimation

Starting from a set of object detections and an estimate of the camera orientation (see 4.3 for more details about orientation computation), our method consists in jointly solving the data association and camera position estimation problems.

In practice, every possible association between detected objects and 3D ellipsoids is determined from label compatibility. As the pose can be computed from one ellipse-ellipsoid correspondence, a pose is computed for each individual pair of ellipse-ellipsoid hypothesis. A consensus set is built at the level of ellipsoids: for each of these potential poses, we reproject all the 3D ellipsoids into the image, and consider ellipse - ellipsoid pairs as inliers when their labels are compatible and the *intersection over union* (IoU) score between the detected and reprojected bounding boxes is greater than a certain threshold (0.5 in our experiments). Note that when a 3D object reprojects onto several 2D detections, only the one with the greatest IoU is considered.When two configurations lead to the same number of inliers, the one with the greatest sum of IoU scores is selected.

4.3 Orientation estimation

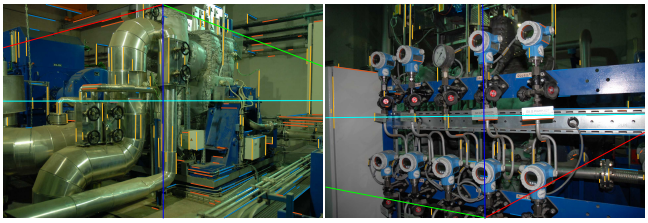

Figure 4: Manhattan vanishing points are reliable features to compute the camera orientation. This figure shows some example results in complex industrial scenes. Those were obtained by using the method described in [31]. Vanishing points are represented by the consistent line segments (one color per vanishing point) and the directions of the detected Manhattan frame are shown in red, green, blue.

In our method, we assume that the camera orientation is known. Having said that, the goal of this section is to briefly describe how this matrix can be obtained in practice, and how our method can provide a mechanism to remove the ambiguity inherent in some orientation determination methods. Inertial Measurement Units (IMUs) are electronic devices whose orientation is measured based on a combination of accelerometers and gyroscopes, sometimes also magnetometers. IMUs are contained in almost all smartphones and tablets, and can easily be used to provide the camera orientation matrix required by our method. Futhermore, this matrix can also be obtained purely from images thanks to image analysis methods based on vanishing point (VP) detection [30, 31, 38]. Indeed, VP detection now performs relatively well in various kinds of environments, including indoor and industrial ones. In the experimental results shown in Section 5.2, we use the method described in [31], since it has proven well suited to various man-made environments.

Using VPs, the camera orientation can be obtained with regard to a frame aligned with three orthogonal directions of the scene that give rise to three specific (so-called Manhattan) VPs. Images of man-made (including industrial) environments often contain such a triplet of orthogonal VPs (see e.g. Fig. 4). It has been many years since the idea of calculating the orientation of the camera from the Manhattan vanishing points was proposed [14], but we address here some issues that are often not considered in the literature while encountered in practice: how to calibrate the rotation between the Manhattan frame and the frame in which the ellipsoids are expressed, and how to deal with the problem of axes exchanges or symmetries between these two frames.

Computing the rotation between the ellipsoids and the Manhattan coordinate frames Ellipsoids are expressed in the world coordinate frame, whose orientation does not necessary fit the Manhattan directions of the scene. For that reason, we need to compute the rotation ${}^{m}R_{w}$ from the world (ellipsoids) frame to the Manhattan frame, and use ${}^{c}R_{w} = {}^{c}R_{m}\,{}^{m}R_{w}$, where ${}^{c}R_{m}$ is the Manhattan-to-camera rotation matrix (obtained using [31]), when computing the position of the camera with the method described in section 3.2. To do this, we detect the Manhattan VPs in N images where the world-to-camera rotations ${}^{c}R_{w}{}^{(i)}$ are known. This provides N matrices ${}^{m}R_{w}{}^{(i)} = {}^{c}R_{m}{}^{(i)\top}\,{}^{c}R_{w}{}^{(i)}$, which should all be the same if data were not corrupted by noise. For better accuracy, we use the orthogonal

projection of $\overline{{}^{m}R_w} = \sum_{i=1}^{N} \frac{{}^{m}R_w^{(i)}}{N}$ onto the special orthogonal group SO(3), given by:

$$ {}^{m}R_w = \overline{{}^{m}R_w} U \left(\frac{1}{\sqrt{\Lambda_1}}, \frac{1}{\sqrt{\Lambda_2}}, \frac{s}{\sqrt{\Lambda_3}} \right) U^{\top}, $$

where $\Lambda_1 \geq \Lambda_2 \geq \Lambda_3 \geq 0$ are the eigenvalues of $M = \overline{{}^{m}R_w}^{\top} \overline{{}^{m}R_w}$, $U \top \text{diag}(\Lambda_1, \Lambda_2, \Lambda_3) U$ is the SVD of M and s the determinant of $\overline{{}^{m}R_w}$ [21].

Dealing with the problem of axes exchanges or symmetries Depending on the position of the camera with regard to the scene, the X and Y axes of the computed Manhattan frame may be exchanged or mirrored with regard to the ones attached to the scene (and used to compute the calibration matrix ${}^{m}R_w$). To tackle this issue, we consider in the RANSAC-like procedure described in section 4.2, each of the four possible cases for the Manhattan-to-camera rotation matrix ${}^{c}R_m$: $\left({}^{c}R_{m:,1}\ {}^{c}R_{m:,2}\ {}^{c}R_{m:,3}\right)$, $\left(-{}^{c}R_{m:,1}\ -{}^{c}R_{m:,2}\ {}^{c}R_{m:,3}\right)$, $\left({}^{c}R_{m:,2}\ -{}^{c}R_{m:,1}\ {}^{c}R_{m:,3}\right)$, $\left(-{}^{c}R_{m:,2}\ {}^{c}R_{m:,1}\ {}^{c}R_{m:,3}\right)$, and keep the one that maximizes the consensus set as explained in Section 4.2.

5 Results

5.1 Pose accuracy with one object

5.1.1 The LINEMOD dataset

We first evaluate our method for estimating the camera position from one object detection in the image on the standard LINEMOD dataset [11]. This dataset is designed to benchmark 6D object pose estimation algorithms, and several accuracy metrics are commonly used: reprojection error, IoU score, ADD metric, ... (see for instance [34] for more details). However, our training-free method based on ellipsoidal modelling of 3D objects and elliptic modelling of their 2D projections is designed for rough camera relocalization instead of accurate pose estimation.

5.1.2 Technical details and results

Most state-of-the-art object detection methods give results in the form of a rectangular bounding box aligned with image axes [10, 18, 26]. To simulate this behaviour, we first project the groundtruth 3D object point cloud into the image using the groundtruth camera projection matrix, and then compute the bounding box of obtained 2D points. The ellipse that inscribes the bounding box is finally used as an approximation of the projected object, as suggested in [3, 28]

We randomly pick 50 frames per object (the dataset contains 15 objects, with roughly 1200 images per each) to build their ellipsoidal models using [28]. All the other frames are used for testing. During the tests, we add a uniform noise lower than a given threshold (0° (no perturbation), 1°, then 2° to the 3 Euler angles (EA) of the groundtruth camera orientation to simulate measurements given by inertial sensors. The overall error on the camera orientation can reach 2° in the first case (1°/EA), and 4.5° in the second case (2°/EA).

The first metric used to evaluate our method is the reprojection error of model points. Usually, estimated poses are considered as correct when the mean reprojection error is lower than a given threshold in pixels (usually 5). Table 1 presents our results on the 15 LINEMOD objects in comparison with the state-of-the-art object pose estimation method [34]. It is important noting that the aims of the two methods are not identical. Indeed, the reference's goal is to accurately estimate the whole camera pose based on object-specific training, whereas our generic method aims at performing rough camera relocalization from object(s) present in the scene, and thus relies on a sometimes rough modeling of objects in the form of ellipsoids. Despite that, our method appears to be fairly accurate (almost every frame presents a mean reprojection error lower than 20 pixels), and is even more accurate than the reference on 23% of the objects (see *eggbox*, *duck*, and *ape*). Moreover, results show that our method is robust to the perturbation applied on camera orientations, since performances do not present a significant decrease when the level of noise increases. More detailed results of our method are provided in Figure 5 (left column), for a level of orientation noise equal to 1°.

Table 1: Comparison of our camera relocalization approach with the state-of-the-art accurate object pose estimation method [34] on the LINEMOD dataset. We report percentages of correctly estimated poses. For our method, we report the results depending on three different levels of perturbation applied on camera orientations, in ° per Euler angle. The **bold face** numbers indicate the best method according to the 5-pixel threshold metric. Note that even if we don't need any object-specific training, our method is fairly accurate, and even outperforms the reference on three objects. Moreover, our method appears to be robust to the perturbation on the camera orientation.

Method	Tekin *et. al.* [34]	Ours					
Perturb.	-	0°	1°	2°	0°	1°	2°
Thresh.	5 pix.	5 pixels			20 pixels		
Object							
ape	92.10	**94.69**	**94.77**	**94.69**	100	100	100
benchvise	**95.06**	12.07	12.16	11.56	93.1	93.1	93.0
bowl	-	**79.31**	**79.05**	**78.46**	100	100	100
cam	**93.24**	49.61	49.44	49.35	100	100	100
can	**97.44**	58.99	58.38	57.77	100	100	100
cat	**97.41**	81.84	81.93	81.58	100	100	100
cup	-	**61.60**	**61.68**	**61.09**	100	100	100
driller	**79.41**	8.79	8.61	8.08	98.4	98.2	98.5
duck	94.65	**94.85**	**94.93**	**94.68**	100	100	100
eggbox	90.33	**97.26**	**97.18**	**96.59**	100	100	99.9
glue	**96.53**	18.96	18.79	18.19	100	100	100
holepunc.	**92.86**	91.41	91.33	91.25	100	100	100
iron	**82.94**	37.11	36.66	35.30	100	100	100
lamp	**76.87**	13.67	12.73	12.39	100	100	99.9
phone	**86.07**	17.85	17.69	17.27	100	99.8	99.9
Average	**90.37**	54.53	54.36	53.88	99.4	99.4	99.4

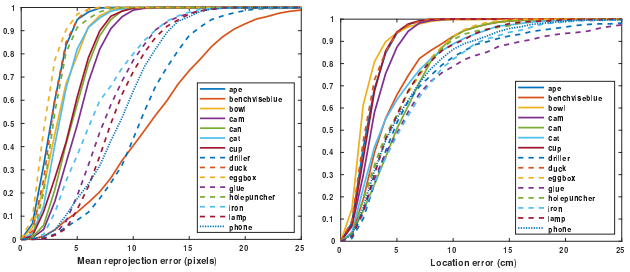

Figure 5: Cumulative density functions of our mean reprojection errors (in pixels) [left] and camera location errors (in cm) [right] on the LINEMOD dataset.

The second metric used to evaluate our method is the 3D pose error. Starting from a noisy orientation obtained by adding a noise of magnitude 1° on each groudtruth Euler angle, the overall camera orientation error do not exceed 2°. The accuracy in terms of position are presented in Figure 5 (right column). Note that maximum diameters of the objects range from 10 cm to 30 cm, and that the average distance between cameras and objects is approximately 92 cm. Considering the top 5 of objects (*bowl*, *duck*, *ape*, *cam*, *cup*), the distance between estimated camera positions and the groundtruth is always lower than 9 cm. In the worst case (*glue*), that distance do not exceed 20cm in 90% of cases. Even if our method can compute the camera position based on a single object detection, it is designed

to take benefit of every object present in the image. That worst-case level of accuracy would thus be reached only in very difficult configurations.

5.1.3 Results interpretation

To provide an in-depth analysis of the previous results, we investigate the effect of the generic ellipsoidal modelling of diverse objects on final camera relocalization performances. Indeed, results presented in 5.1.2 show major differences depending on the object considered for testing. Our method relies on the detection of a virtual ellipse considered as projection of the 3D object model (ellipsoid). As a consequence, it highly depends on our ability to detect ellipses similar to the groundtruth projection of the model. To quantify the gap between effective and expected detections, we define a detection error as the average distance between the 4 vertices (endpoints of principal axes) of the detected ellipse and their closest points on the ellipse projected with groundtruth camera. Figure 6 shows the correlation between the mean reprojection error on the whole LINEMOD dataset (illustrating our relocalization performance) and the original detection error. Figure 7 illustrates more concretely that phenomenon on the best-case (*eggbox*) and worst-case (*driller*) objects.

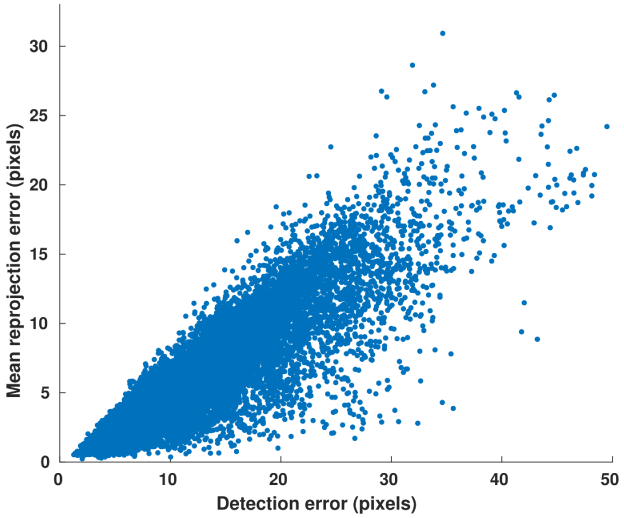

Figure 6: Mean reprojection errors (in pixels) of our method on the whole LINEMOD dataset, with respect to ellipse detection errors (in pixels). That figure shows the correlation between our relocalization performance and the original error on ellipse detection.

5.2 Real-life scenarios

5.2.1 The RGB-D TUM dataset

We now evaluate the robustness of the full camera relocalization method on the standard RGB-D TUM dataset [33]. Even if this dataset was originately created to benchmark SLAM algorithms, its sequences containing repeated common objects and occlusions make it of interest to assess our algorithm performances.

We use two sequences for testing: *fr2/desk* and *fr3/long_office*. Both of them represent office environments with repeated objects such as computers, books, cups, or bottles. They are composed of approximately 2700 images taken by a standing person performing a closed loop around a central desk. The scenes and operator trajectories are roughly contained in a square with 4-meter side in *fr2/desk*, and 5-meter side in *fr3/long_office*.

The first scene *fr2/desk* is composed of 16 objects grouped in 11 categories (up to 3 objects per category). In total, 104 images were used to build the model, and the remaining 2861 for testing. The second scene (*fr3/long_office*) is composed of 28 objects with 9 different labels (up to 10 objects with the same label). Among the 2559 images of the sequence, 71 were necessary to reconstruct the ellipsoids.

In our experiments, only RGB images are used (no depth information). The ground truth data was obtained by motion capture, but some cameras are provided without ground truth (705 in *fr2/desk* and 2 in *fr3/long_office*). The camera intrinsic parameters are known.

5.2.2 Orientation estimation

Our method is assessed using IMU-simulated orientations as well as orientations obtained from vanishing point detection.

IMU orientations are simulated by adding an uniform noise between $-1°$ and $+1°$ to each Euler angle of the groundtruth camera orientation, leading to an overall error of at most $2°$.

Manhattan vanishing points were obtained using the procedure described in Section 4.3. In order to compute the world-to-Manhattan calibration rotation ${}^{m}R_{w}$, we considered the same images as those used to reconstruct the ellipsoids, and kept the ones that allowed us to detect the Manhattan frame (see Section 4.3). Finally, two images were usable with *fr2/desk* and four with *fr3/long_office*. We obtained, respectively:

$$ {}^{m}R_{w} = \left(\begin{array}{c|c|c} 0.9994 & -0.0141 & 0.0212 \\ 0.0141 & 0.9996 & -0.0202 \\ -0.0202 & 0.0208 & 0.9993 \end{array}\right), {}^{m}R_{w} = \left(\begin{array}{c|c|c} 1.0000 & -0.0033 & 0.0062 \\ 0.0034 & 0.9999 & -0.0126 \\ -0.0060 & 0.0125 & 0.9998 \end{array}\right). $$

These matrices are very close to the identity matrix, which means that the world frame was already aligned with the Manhattan frame (defined by the borders of the desks) when using the ground truth poses to reconstruct the ellipsoids.

fr2/desk and *fr3/long_office* are actually hard sequences in terms of VP detection. Figure 10, last column shows the ground truth paths followed by the camera (projection of the camera positions onto the ground plane, plotted in black) in these two sequences. Parts of the paths where the Manhattan VPs could not be obtained are indicated by purple curves, and typical sources of failures are shown in Fig. 8, using the same numbers as for the path parts. In part 1, failures or inaccuracies are often due to the fact that the screen, that is slightly tilted, occupies a large part of the image, which misleads the detection of the zenith. In part 2, a plant occludes a large part of the scene, which misleads the detection of the horizontal VPs. In part 3, the vertical book generates several lines segments that correspond to near-vertical, parallel lines in the scene that, again, mislead the detection of the zenith. Moreover, the legs of the chairs have a star shape, leading to several line segments that meet at the center of the legs, which produces spurious vanishing points. In part 4, the Teddy bear occupies a large part of the image. Outside of these difficult cases, the Manhattan frame is regularly found all along the paths.

5.2.3 Detection and matching

In our experiments, object detection is performed by using YOLOv3 [26]. Globally, the procedure described in section 4.2 has enough inliers to robustly compute the pose, and one inlier can even be sufficient. Some parts of the sequences, however, present particular difficulties that we now describe. These parts are indicated by orange curves in Figure 10 (first column) and illustrated by typical images in Figure 9, using the same indices in both figures.

On *fr2/desk*, some parts of the sequence exhibit a very few (or even none) fully visible objects. In practice, images that do not contain detections, or whose detected labels are not present in the model, are ignored. In other cases, the very few correct detections can suffer from severe occlusions and/or truncature (see indices 1,2 and 5), and/or be coupled with false detections (see indices 3 and 4), leading to failure in camera relocalization.

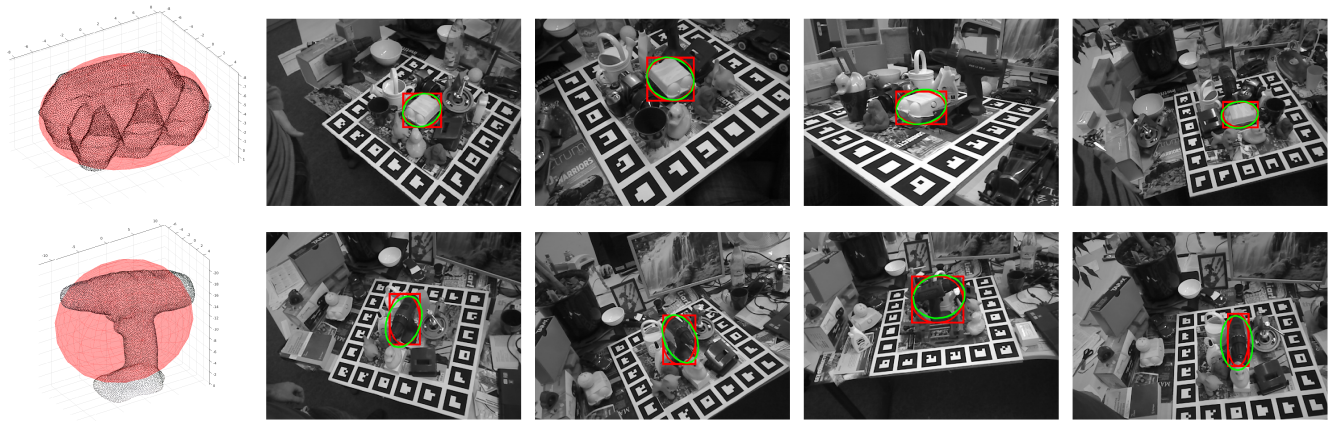

Figure 7: Illustration of the error on ellipses inscribed in detected bounding boxes (red), with respect to ellipse projected with groundtruth camera (green). That error is much more important on the LINEMOD *driller* object (row 2) which cannot be accurately modelled by an ellipsoid, than on the *eggbox* object (row 1) which is closely fitted by the reconstructed ellipsoid. The first column shows groundtruth 3D point clouds (black) and corresponding reconstructed ellipsoids (red).

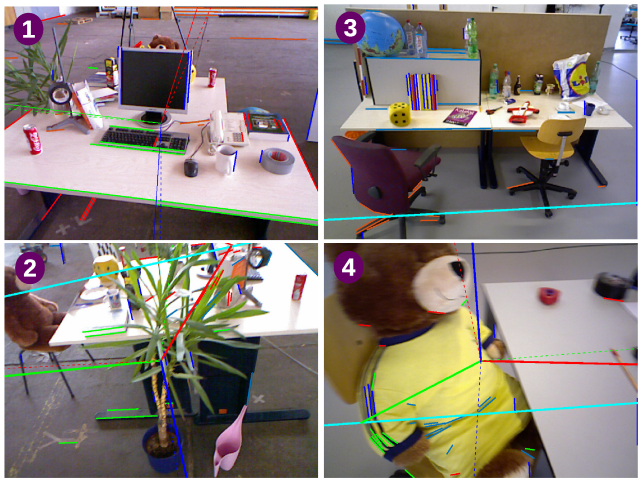

Figure 8: Failure cases of Manhattan vanishing point detection. The computed Manhattan frame is shown in red, green, blue solid lines, the ground true one in dashed lines.

On *fr3/long_office*, the most problematic part is referred as (6). It consists in images in which a single chair is detected, whereas three different chair instances are present in the model. Considering a single detection with label ambiguity, and without any precise geometric information about reconstructed objects (only rough ellipsoids), our method sometimes fails to associate the detection to the object it originates from.

5.2.4 Quantitative results

Pose relocalization is assessed on *fr2/desk* and *fr3/long_office* using both IMU simulated orientations and orientations from VPs. For comparison purpose, we also tried to compute the full six parameters of the pose by iteratively minimizing the geometric projection error of [22] defined in equation (1) (second term, with $\mathbf{q}_j$ fixed and no attachment to odometry). In that case, we used the Levenberg Marquardt algorithm to perform the optimization and the ground truth data to initialize the pose parameters. For the sake of equity, we used the inlier correspondences provided by our method.

Table 2 shows the mean and median location and orientation errors obtained by the three procedures on both sequences. All

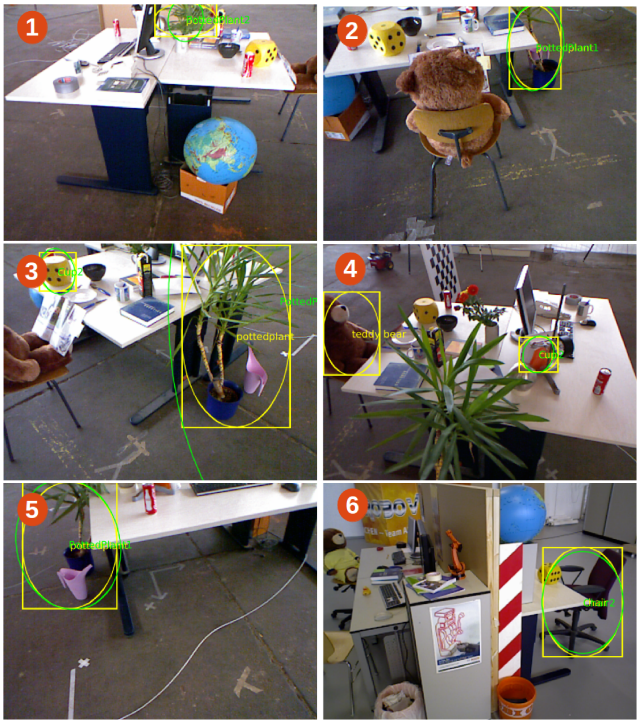

Figure 9: Failure cases of our method. (1, 2, 5): only one truncated object is detected, leading to badly estimated size of the projection. (3,4): only two detections and one is false (*cup*), misleading our algorithm. (6): only one detection with ambiguous label (*chair*). In this last case, our algorithm associates the detection to the model chair #2 instead of #1.

computed locations are shown projected onto the ground plane in Figure 10. In these figures, the procedure with simulated orientations is referred to as *IMU*, the procedure based on VPs as *VPs* and the procedure minimizing the geometric error as *geomQS*.

IMU-simulated orientations Camera poses obtained with simulated orientations are relatively close to the ground truth path in both sequences (Figure 10, first column), though the camera point cloud is slighlty more chaotic in *fr2/desk*. This is due to the fact

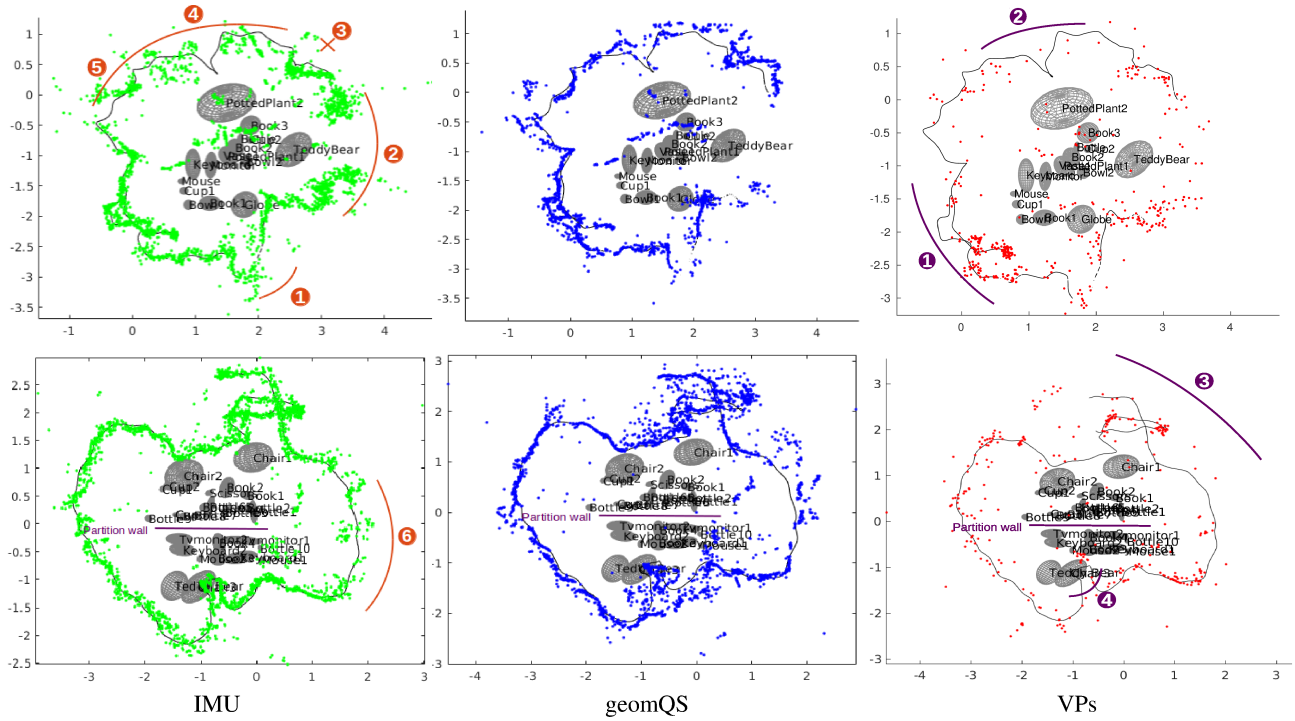

Figure 10: Results of camera relocalization with our method and IMU orientations (column 1), 6-DOF optimization (column 2) and our method with orientations from VPs (column 3). The ground truth path is shown in black solid line, the camera positions (projected onto the ground plane) with colored dots. First row: *fr2/desk* Second row: *fr3/long_office*.

Table 2: Errors on the RGB-D TUM dataset in terms of camera location (*loc.*, in m) and orientation (*ori.*, in °). In the table, *italic* numbers denote synthetic data (noise added to the ground truth).

Dataset		*fr2/desk*			*fr3/long_office*		
Method		IMU	geomQS	VPs	IMU	geomQS	VPs
loc.	avg	0.33	0.33	0.42	0.16	0.28	0.93
	std	0.51	0.48	0.79	0.29	0.35	1.42
	med	0.11	0.12	0.13	0.08	0.14	0.14
ori.	avg	*0.95*	1.3	16.7	*0.93*	3.8	35
	std	*0.32*	3.5	47.0	*0.3*	7.7	65
	med	*0.94*	0.03	2.5	*0.92*	0.04	1.7

that this sequence contains more challenging images in terms of detection, as explained in previous section. In *fr3/long_office*, only one part of the sequence is particularly difficult to manage, due to the fact that only one chair is detected while several are contained in the model. Despite these diffculties, our method with IMU simulated data presents excellent relocalization performances: the median error on the camera location is 8cm on *fr3/long_office* and 11cm on *fr2/desk*.

Orientations from VPs When using the Manhattan VPs to compute the camera orientation, it can be seen that the camera point cloud is much sparser than when using the IMU simulated data (Figure 10, last column). Of course, this approach can only succeed if both object detection and the Manhattan VP detection succeed, and therefore inherits all types of problems that may occur for each procedure. For that reason, the mean errors shown in Table 2 are high, and especially for *fr3/long_office*. However, the median errors (13cm/2.5° for *fr2/desk*, 14cm/1.7° for *fr3/long_office*) are relevant to relocalization.

6-DOF optimization of the geometric error The camera point cloud obtained by 6-DOF optimization of the geometric error is relatively unstable (Figure 10, middle column): half of the time, the computed pose is very close to the ground truth, but half of the time it diverges from it. This can also be deduced from the mean errors on position and orientation in Table 2, that are large, while the median errors are very small. We analyse this result as follows. First, 6-DOF optimization can more easily compensate for accuracy errors obtained on ellipses (fitted to the BBs) than when the orientation is constrained. Second, the geometric error tends to favour larger ellipses, to the detriment of small objects as illustrated, for example, in Figure 11, *fr2/desk*-f1. This confirms that this method requires an attachment to odometry (first term in equation (1)). It must also be noticed that, for that reason, we prefer not to follow our closed-form computation with a non-linear optimization of the pose as is commonly done in classical pose computation.

Finally, Figure 11 qualitatively illustrates the robustness of our method on both suitable (*fr2/desk*-f1, *fr3/long_office*-f1) and challenging configurations : repeated patterns (*fr3/long_office*-f1,f4,f5), single object detection (*fr2/desk*-f2, *fr3/long_office*-f2), occlusions (*fr2/desk*-f1,f2,f3) motion blur (*fr3/long_office*-f3) and various camera-scene distances and relative orientations.

5.3 Discussion and potential applications

In this section, we discuss the advantages and disadvantages of our object-based relocalization method in comparison with low-level methods based on local features. By nature, our method focuses on semantically relevant features in the images of the scene, which leads us to deal with light 3D models encoding only the semantics of the scene and leaving out unnecessary information. Moreover, it operates at the level of objects, which are inherently more discriminative than local features in scene understanding and localization tasks, and are also more robust to viewpoint or illumination changes [20].

At the opposite, low-level methods have the advantage to be context-independent, in the sense that they can be applied on images of any environments, as soon as images are sufficiently textured.

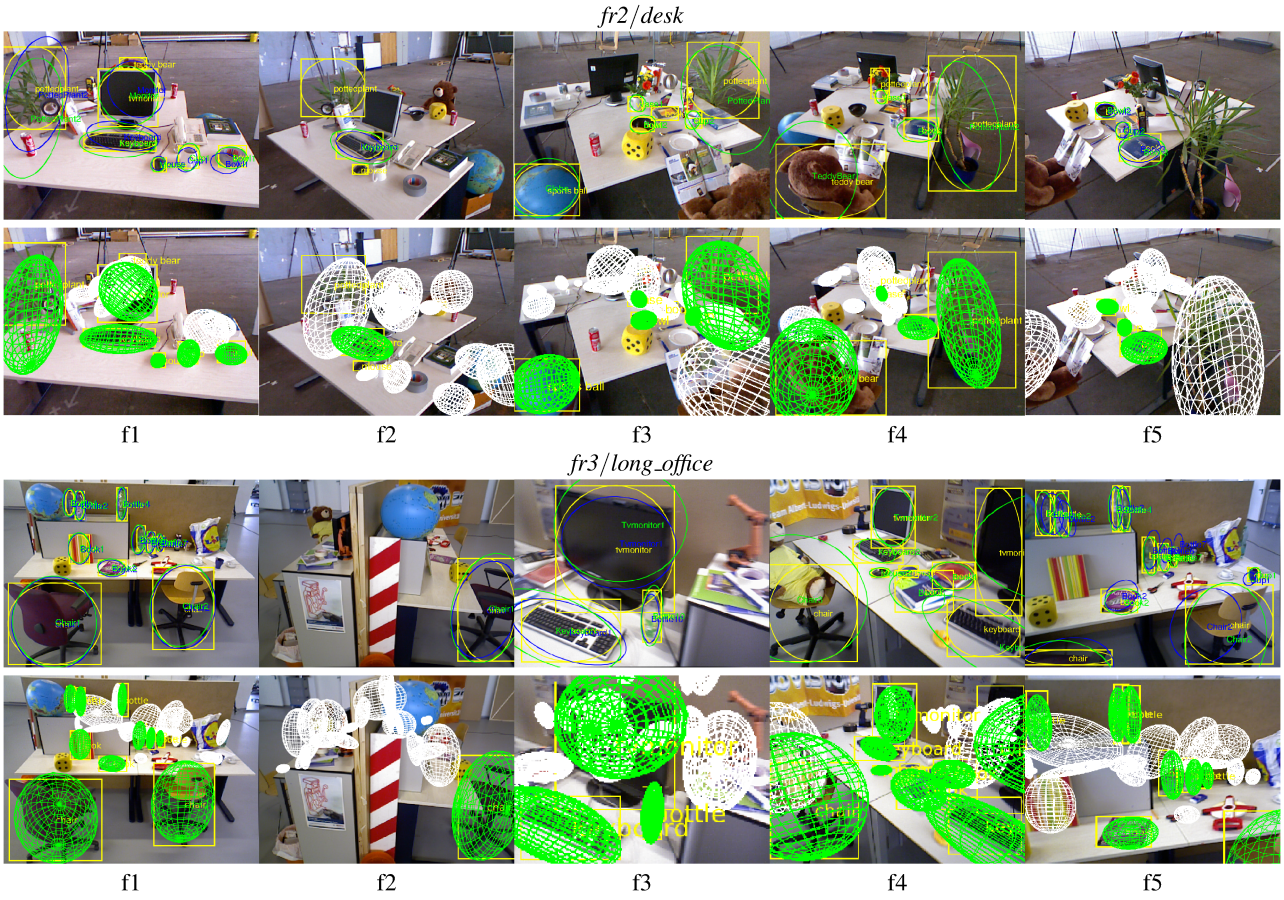

Figure 11: Illustration of the robustness of our method on several images from the RGB-D TUM dataset (rows 1-2: sequence *fr2/desk*, rows 3-4: sequence *fr3/long_office*). Detected bounding boxes and inscribed ellipses are displayed in yellow, with automatically generated label. Reprojections of inlier ellipsoids are displayed in green, whereas the other ones are displayed in white. Note that objects cannot be classified as inliers if there projections are not detected in the image. In rows 1-3, ellipses reprojected using the pose estimated by *geomQS* are displayed in blue.

On the contrary, our method is dependent on the object detection algorithm, thus can deal only with object classes learned by the detector. Applying our method on a new environment thus requires to train the detector on objects mainly encountered in this kind of environment.

Finally, the feature-based and object-based models each have their own limitations. In the context of indoor or industrial applications, large parts of the scene are untextured. With Structure from Motion techniques, features can be clustered in small regions of the images, leading to unstable pose estimation. In addition repeated local features are most often simply discarded because of their ambiguity and do not appear in the final scene model [12, 32]. Considering object-based models allow to consider wider and untextured areas of the scene for pose computation, allowing greater robustness, albeit at the expense of accuracy due to the approximation of objects as ellipsoids.

A typical use of our method is illustrated in Fig. 12 : an operator moves freely in a factory, equipped with AR glasses. The pose of the glasses is calculated using our system (Fig. 12-bottom) and information such as the ones shown in Fig. 12-top are displayed in AR to help him follow a planned route and perform maintenance or training tasks. Industrial environments generally contain many objects from small (e.g. a valve) to large (e.g. an equipment) that can be used as object landmarks with our system. When a CAD model of an object is available, more accurate 3D overlays can be obtained if necessary, using any 3D-2D registration method (e.g. [1] for industrial objects). Such methods require an initial estimate of the pose that our method can provide. Many other scenarios can be considered in various fields such as sales (AR in shops), entertainment or museography, as long as enough objects are present in the environment, on which our system can rely.

6 Conclusion

In this paper, we explored means to perform relocalization at the level of object. We took advantage of progress realized in object detection which allows us to generate 2d/3d correspondences between object detected in images and approximated by an ellipse, and 3D objects represented by ellipsoids. Assuming that an estimate of the camera rotation is available, we proposed a closed form method to compute the camera localization from one ellipse-ellipsoid correspondence. A RANSAC-like algorithm operating at the level of object is proposed to cope with wrong data associations either due to erroneous object detection or to the presence of repeated objects in the scene. The conducted experiments proved the effectiveness of the method even when a small number of objects are detected. As shown in the experiments, rotation information provided by IMU or vanishing point detection turns out to be sufficient for relocalization. Accurate iterative model-based methods can then be used from this first estimate to refine the estimate of the pose.

The method has many advantages. By considering pose com-

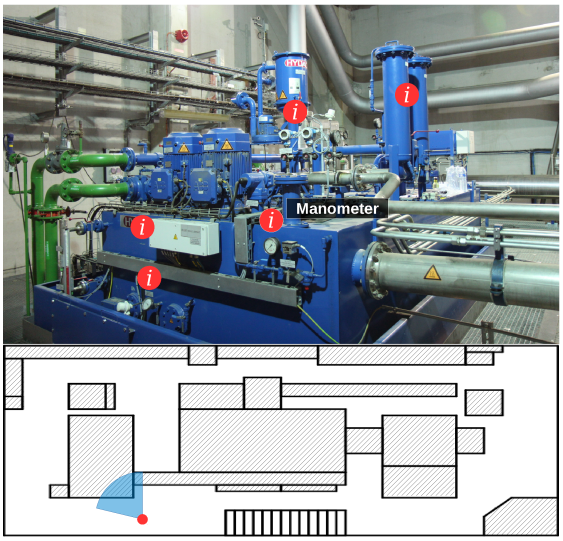

Figure 12: Illustration of a typical use-case of our system. The operator is automatically localized with respect to the devices of interest, and information for maintenance tasks are added in AR.

putation at the level of objects, we avoid common problems due to repeated patterns encountered with feature-based methods in man made environments. In addition , the combinatory of ellipse-ellipsoid correspondences is relatively small, which opens the way towards efficient relocalization in large environments, where only the prominent objects are integrated in the model. Future works will be dedicated to experiments of the method for maintenance in industrial scenes.

APPENDIX A: PROOF OF THEOREM 1

Let B' be the backprojection cone. B' is thus a real, symmetric and invertible matrix of signature (1,2) or (2,1). As shown in eq (3), if the ellipsoid projects onto the ellipse, then

$$A\mathbf{\Delta} = \sigma B'\mathbf{\Delta}$$

The possible solutions for σ are generalized eigenvalues of $\{A,B'\}$ and are thus roots of the *generalized characteristic polynomial* $P_{\{A,B'\}}(x) = det(A - xB')$.

Since A is positive definite and B' is symmetric, the couple $\{A,B'\}$ has the following properties [8]:

1. the generalized eigenvalues are real,
2. the reducing subspaces are of the same dimension as the multiplicity of the associated eigenvalues,
3. the generalized eigenvectors form a basis of $\mathbb{R}^3$, and those with distinct eigenvalues are A-orthogonal.

Moreover, since B' is invertible, we can easily notice that the generalized eigen elements of $\{A,B'\}$ are the same as the eigen elements of $B'^{-1}A$. We can then observe that $Q(x) = \mu x^2 - (\mu+1)\sigma x + \sigma^2$, where $\mu = 1 - \mathbf{\Delta}^\top A\mathbf{\Delta}$, is an annihilator polynomial of $B'^{-1}A$ (see Appendix B). Since Q is of degree 2, we can infer that $B'^{-1}A$, and thus $\{A,B'\}$, have at most two distinct eigenvalues.
However, the case of one eigenvalue of multiplicity 3 denoted σ_0 is impossible. Indeed, according to property 2) above, this will imply $dim(Ker(A - \sigma_0 B')) = 3$, *i.e.* $A = \sigma_0 B'$, which is impossible because A represents an ellipsoid while B' represents a cone. So **the couple has exactly two distinct generalized eigenvalues**.This concludes the proof of the first item of Theorem 1.

Let us now denote σ_1,σ_2 these eigenvalues and $\mathbf{\Delta}_1$, $\mathbf{\Delta}_2$ two associated eigenvectors, such that σ_i is the eigenvalue of multiplicity i and $\|\mathbf{\Delta}_i\| = 1$. Let's suppose now that there is $k \in \mathbb{R}^*$ such that $(A, \sigma_2, k\mathbf{\Delta}_2)$ is solution of (1). We therefore have :

$$A - \sigma_2 B' = MA$$

where $M = k^2(\mathbf{\Delta}_2^\top A\mathbf{\Delta}_2 I - A\mathbf{\Delta}_2\mathbf{\Delta}_2^\top)$ and I is the identity matrix. According to the property 2 (above), $dim(Ker(A - \sigma_2 B')) = 2$. Since A is invertible, $dim(Ker(M)) = dim(Ker(AM)) = 2$. However, we observe that

$$\begin{aligned}\forall \mathbf{X} \perp \mathbf{\Delta}_2,\ M\mathbf{X} &= k^2\mathbf{\Delta}_2^\top A\mathbf{\Delta}_2\mathbf{X} - k^2 A\mathbf{\Delta}_2\mathbf{\Delta}_2^\top\mathbf{X} \\ &= k^2\mathbf{\Delta}_2^\top A\mathbf{\Delta}_2\mathbf{X} - k^2 A\mathbf{\Delta}_2(\mathbf{\Delta}_2\cdot\mathbf{X}) \\ &= k^2\mathbf{\Delta}_2^\top A\mathbf{\Delta}_2\mathbf{X}\end{aligned}$$

Since A is positive definite, $\mathbf{\Delta}_2^\top A\mathbf{\Delta}_2 > 0$. This implies that $M\mathbf{X} \neq 0$ when $\mathbf{X} \in \mathbf{\Delta}_2^\perp$ and that $\mathbf{\Delta}_2^\perp \cap Ker(M) = \{0\}$. This is a contradiction since $\mathbf{\Delta}_2^\perp$ and $Ker(M)$ are two subspaces of $\mathbb{R}^3$ of dimension 2. As a result, the only possible value σ is the eigenvalue of multiplicity 1.

APPENDIX B: $Q(B'^{-1}A) = 0$

Replacing (3) into (2), we obtain:

$$\sigma^2 B'\mathbf{\Delta}\mathbf{\Delta}^\top B' - (\sigma\mathbf{\Delta}^\top B'\mathbf{\Delta} - 1)A = \sigma B'$$

We can then deduce the following expression for A:

$$A = \frac{\sigma}{1 - \sigma\mathbf{\Delta}^\top B'\mathbf{\Delta}}(B' - \sigma B'\mathbf{\Delta}\mathbf{\Delta}^\top B')$$

Thus, denoting I the identity matrix and defining $f = \frac{\sigma}{1-\sigma\mathbf{\Delta}^\top B'\mathbf{\Delta}}$, then left-multiplying by B'^{-1}, we obtain

$$B'^{-1}A = f(I - \sigma\mathbf{\Delta}\mathbf{\Delta}^\top B')$$

Squaring that expression leads to

$$\begin{aligned}(B'^{-1}A)^2 &= f^2(I - \sigma\mathbf{\Delta}\mathbf{\Delta}^\top B')^2 \\ &= f^2(I - 2\sigma\mathbf{\Delta}\mathbf{\Delta}^\top B' + \sigma^2\mathbf{\Delta}(\mathbf{\Delta}^\top B'\mathbf{\Delta})\mathbf{\Delta}^\top B') \\ &= f^2(I - 2\sigma\mathbf{\Delta}\mathbf{\Delta}^\top B' + \sigma^2(\mathbf{\Delta}^\top B'\mathbf{\Delta})\mathbf{\Delta}\mathbf{\Delta}^\top B') \\ &= f^2(I - \sigma(2 - \sigma\mathbf{\Delta}^\top B'\mathbf{\Delta})\mathbf{\Delta}\mathbf{\Delta}^\top B')\end{aligned}$$

Defining $\mu = 1 - \sigma\mathbf{\Delta}^\top B'\mathbf{\Delta} = 1 - \mathbf{\Delta}^\top A\mathbf{\Delta}$:

$$\begin{aligned}(B'^{-1}A)^2 &= f^2(I - \sigma(\mu+1)\mathbf{\Delta}\mathbf{\Delta}^\top B') \\ &= f^2((\mu+1)(I - \sigma\mathbf{\Delta}\mathbf{\Delta}^\top B') - \mu I) \\ &= f(\mu+1)B'^{-1}A - f^2\mu I \\ &= \frac{\sigma}{\mu}(\mu+1)B'^{-1}A - \frac{\sigma^2}{\mu}I\end{aligned}$$

Finally, we have

$$\mu(B'^{-1}A)^2 = \sigma(\mu+1)B'^{-1}A - \sigma^2 I$$

Thus, denoting $Q(x) = \mu x^2 - (\mu+1)\sigma x + \sigma^2$,

$$Q(B'^{-1}A) = 0$$

REFERENCES

[1] B. Besbes, S. N. Collette, M. Tamaazousti, S. Bourgeois, and V. Gay-Bellile. An interactive augmented reality system: A prototype for industrial maintenance training applications. In *ISMAR*, 2012.

[2] A. Crivellaro, M. Rad, Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit. A novel representation of parts for accurate 3d object detection and tracking in monocular images. In *ICCV*, 2015.

[3] M. Crocco, C. Rubino, and A. Del Bue. Structure from motion with objects. In *CVPR*, 2016.

[4] D. Eberly. Reconstructing an ellipsoid from its perspective projection onto a plane. `https://www.geometrictools.com/`, May 2007. Updated version: March 1, 2008.

[5] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[6] P. Gay, V. Bansal, C. Rubino, and A. Del Bue. Probabilistic structure from motion with objects (psfmo). In *ICCV*, 2017.

[7] R. B. Girshick. Fast R-CNN. In *ICCV*, 2015.

[8] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third ed., 1996.

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second ed., 2004.

[10] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *ICCV*, 2017.

[11] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *ACCV*, 2012.

[12] E. Hsiao, A. Collet, and M. Hebert. Making specific features less discriminative to improve point-based 3d object recognition. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pp. 2653–2660, 2010.

[13] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. SSD-6D: making rgb-based 3d detection and 6d pose estimation great again. In *ICCV*, 2017.

[14] J. Kosecka and W. Zhang. Video compass. In *ECCV*, 2002.

[15] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate $O(n)$ solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155–166, 2009.

[16] J. Li, D. Meger, and G. Dudek. Context-coherent scenes of objects for camera pose estimation. In *IROS*, 2017.

[17] J. Li, Z. Xu, D. Meger, and G. Dudek. Semantic scene models for visual localization under large viewpoint changes. In *CRV*, May 2018.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *ECCV*, 2016.

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004.

[20] S. M. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. D. Cox, P. I. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Trans. Robotics*, 32(1):1–19, 2016.

[21] M. Moakher. Means and averaging in the group of rotations. *SIAM J. Matrix Analysis Applications*, 24(1):1–16, 2002.

[22] L. Nicholson, M. Milford, and N. Sünderhauf. QuadricSLAM: Dual quadrics from object detections as landmarks in object-oriented slam. *IEEE Robotics and Automation Letters*, 4(1):1–8, Jan 2019.

[23] M. Oberweger, M. Rad, and V. Lepetit. Making Deep Heatmaps Robust to Partial Occlusions for 3D Object Pose Estimation. In *ECCV*, 2018.

[24] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[25] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. In *CVPR*, 2017.

[26] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. `http://arxiv.org/abs/1804.02767`, 2018.

[27] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[28] C. Rubino, M. Crocco, and A. D. Bue. 3d object localisation from multi-view image detections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1281–1294, Jun 2018.

[29] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *CVPR*, 2017.

[30] G. Simon, A. Fond, and M.-O. Berger. A Simple and Effective Method to Detect Orthogonal Vanishing Points in Uncalibrated Images of Man-Made Environments. In *Eurographics*, 2016.

[31] G. Simon, A. Fond, and M.-O. Berger. A-Contrario Horizon-First Vanishing Point Detection Using Second-Order Grouping Laws. In *ECCV*, 2018.

[32] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[33] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *IROS*, 2012.

[34] B. Tekin, S. N. Sinha, and P. Fua. Real-time seamless single shot 6d object pose prediction. In *CVPR*, 2018.

[35] D. S. Wokes and P. L. Palmer. Autonomous pose determination of a passive target through spheroid modelling. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, Aug 2008.

[36] D. S. Wokes and P. L. Palmer. Perspective reconstruction of a spheroid from an image plane ellipse. *International Journal of Computer Vision*, 90(3):369–379, 2010.

[37] K. M. Yi, E. Trulls Fortuny, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform. *ECCV*, 2016.

[38] M. Zhai, S. Workman, and N. Jacobs. Detecting vanishing points using global image context in a non-manhattan world. In *CVPR*, 2016.